Improving Accuracy of the DrQA Model Using Named Entity Recognition and ELMo Embeddings in the Cross-Domain Context

Amit Joshi (*amitjoshi@utexas.edu*), Rithvik Saravanan (*rsaravanan@utexas.edu*)
CS 378: Natural Language Processing

Abstract

Many natural language models for question answering take naive approaches to narrowing down important sections of text. We present a dual approach to remedying this issue that significantly outperforms prior models on retrieving the correct answer from a passage for a particular question. The first component of our dual approach is using Named Entity Recognition to allow our model to better focus on sections of text that contain the correct answer, and simultaneously ignore sections of text that do not contain the correct answer (Mollá et al.). The second component of our modification is to train our model with deep-contextualized word embeddings using ELMo (Iyyer et al.) as opposed to the GLoVE embeddings that the current DrQA model uses (Chen et al.). These approaches resulted in notable, mostly positive, changes to EM and F1. One trade-off of this approach is that the training time is longer due to taking a newer approach of embedding the text. This issue was resolved by training the model on a GPU.

1 Introduction

Previous natural language models we have encountered for the problem of question answering typically use LSTM neural networks to identify the most likely probabilities for the start and end of the answer span in each passage to answer a specific question. For our project, we explored different approaches to modify this spanning algorithm as well as the model's neural embedding architecture for the robustness of the model.

To focus on the spanning algorithm, we first analyzed how identifying named entities in passages would affect the robustness of our model. Named entities can be typical nouns and pronouns, for example, like person names, location names, events, dates, times, and quantities. The process of identifying named entities in text is known as Named Entity Recognition (NER) and is a subset of the topic of information extraction. NER allows us to identify potential correlated entities in both the question and the passage that could affect the model outcomes and answer spans. The idea behind using Named Entity Recognition is that the same named entities can appear in the question and the answer portion of the passage.

In terms of the model's neural architecture, we explored using ELMo embeddings in our LSTM in place of the GloVe pre-trained word embeddings. ELMo is a deep contextualized word representation that is useful for modeling the syntax and semantics of different words as well as how these use cases vary across linguistic contexts. ELMo embeddings are typically applied for use in question answering, textual entailment, and sentiment analysis models.

These techniques are particularly useful in cross-domain settings because they introduce new question and passage types with different words that may not appear in the training set. Since the new words are not as easily recognized by the model, NER is beneficial because it can identify word types and match up words from the question with relatively similar words in the passage. Accordingly, we explored the viability of our model in a cross-domain setting by training on the SQuAD training set and testing it on the NewsQA dev set.

2 Method

2.1 Named Entity Recognition

We will discuss the role of Named Entity Recognition (NER) in Question Answering. Named Entity Recognition is a way of parsing text that extracts the named entities it references with appropriate type tags. Examples of the common tags found by NER are shown in the table below (spaCy).

Looking at the training sample below, we considered how we could leverage discovering named entities within the passages and questions. Once we identified the named entities in the passage based on the question, we strategically modified the probabilities by increasing the likelihood of those specific named entities that showed an association with the question. In doing so, we increased the probabilities of those specific named entities and, by consequence, their adjacent contexts of falling within the answer span.

```
[CONTEXT]
architecturally , the school has a
catholic character . atop the main
building 's gold dome is a
golden statue of the virgin mary .
immediately in front of the main
building and facing it , is a
copper statue of christ with arms
upraised with the legend " venite ad
me omnes " . next to the main
building is the basilica of the
sacred heart . immediately behind
the basilica is the grotto , a
marian place of prayer and
reflection . it is a replica of the
grotto at lourdes , france where the
virgin mary reputedly appeared to
saint bernadette soubirous in 1858 .
at the end
of the main drive ( and in a direct
line that connects through 3 statues
and the gold dome ) , is a
simple , modern stone statue of mary
.

[QUESTION]
to whom did the virgin mary
allegedly appear in 1858 in lourdes
france ?

[ANSWER]
saint bernadette soubirous
```

First, we noticed how we could utilize the fact that a question contains the word “whom”; obviously the answer must be a person or similar entity (organization, country, etc). We thought about how we could map different question keywords to various types of named entities, and increase the probability scores of these named entities. These modified probability scores are used in the spanning algorithm to more accurately narrow down the start and end pointers of the area of the passage where the question lies.

Next, we noticed that in this case, the relevant area of the passage had a lot of named entities in common with the question, such as “virgin mary,” “1858,” “lourdes france” etc. Similarly, we thought it would be wise to increase the probability scores of these named entities to more accurately narrow down the start and end pointers of the area of the passage where the question lies.

Another reason we used NER was due to the fact that we are trying to optimize for cross-domain performance. Specifically for adversarial examples, where niche or unseen vocabulary might be commonplace, it would help to have NER increasing probability scores of words within the passage to narrow down where the answer lies. This is because the embeddings generated for niche or unseen vocabulary may not be the most accurate, given that the model may not have been trained on these rare words.

2.1.1 Named Entity Matching

The reason we looked into Named Entity Matching was because we had a hypothesis that Named Entities in the question often show up in the answer. We imagined that if the question was “How much did Apple sell the iPhone 7 for when it first came out”, the relevant part of the passage would be “Apple sold the iPhone 7 for $600 in 2016.” We knew that increasing the probability of the named entity “Apple” would increase the probability of the start pointer being put on the index of the word “Apple.”

Correctly capturing these named entities allows us to better select areas of text in the passage that might contain the answer. Currently, the implementation only finds an optimal answer span given the start and end probabilities by identifying the maximum joint probability of the answer span for tokens lying a fixed distance away from the optimal starting point.

Initially, we considered how to calculate this in a more nuanced, insightful, and effective manner. To do this, we decided to find the named entities in both the question and the passage using the spaCy Python NLP library. We found that spaCy doesn’t find the named entities well in lower-cased text.

To resolve this issue, we utilized spaCy on the true-case version of the passage in order to better find entities in the text. Given this, we scanned through all the named entities in the passage. For each named entity, if it was found in the question, then the appropriate word index where this named entity was found in the passage had its joint probability score incremented in the logits/probabilities that were returned.

This process was done for both the start and the end pointer in order to handle cases where the named entity occurred before and after the answer. This process was done for each named entity. In addition, if a named entity contained multiple words, all word indices were given incremented scores. We increased each matching named entity’s probability while calculating start/end pointers by 0.5, as larger values hurt performance and smaller values didn’t make any significant difference.

We verified the correctness of our implementation by checking the named entities in both the passage and the question as well as the appropriate corresponding entities and ensuring that the correct indices were located so we could increment that word index's score.

2.1.2 Named Entity Mapping

The motivating factor behind using Named Entity Mapping is that we thought we could figure out what type of answer the question was looking for based on certain keywords. For example, if the question was "Who was the first president of America?" and the relevant part of the passage was "George Washington was the first president of America," then the fact that the question had the word "who" in it would mean that the answer must be a person.

We then explored how we could capture what kind of named entity the question itself is asking for. Depending on certain question keywords in the question, we gave higher probability scores to specific types of named entities in the passage. These question keywords are questioning words that strongly indicate that they are looking for a specific type of named entity. In turn, these higher probability scores are used in the spanning algorithm of the model, which calculates the start and end points in the passage where the model thinks the answer is located. For Named Entity Mapping, we increased each mapped named entity's probability while calculating start/end pointers by 1, as larger values hurt performance and smaller values didn't make any significant difference. We list our question keywords along with their corresponding named entities in the table below.

Question Keyword	**Corresponding Named Entities**
who/whom	"PERSON", "ORG", "GPE", "NORP"
when	"DATE", "TIME"
how much/ how many	"MONEY", "QUANTITY", "PERCENT", "CARDINAL"
where	"FAC", "ORG", "GPE", "LOC", "EVENT"

2.2 ELMo Embeddings

The motivating factor behind using ELMo embeddings is that they can help with the model's understanding of differentiating between homonyms. For example, the phrases "I would like to go to the ball" and "I play with a tennis ball" both use the term "ball". However, in each of these phrases, the meaning of the word "ball" is different. In the first phrase, it is a noun that represents a location/event. In the second phrase, it is a noun that represents an object. Since there are several words in the English language that have multiple meanings, ELMo embeddings is able to improve a model's ability to detect such distinctions using contextualized representations for each word.

In other words, each word embedding keeps track of the context in which each word is used. This contrasts with the original GloVe embeddings that our model previously used because each word had a constant word embedding regardless of its multiple potential meanings (Peters et al.).

2.3 Training Process

Since parsing named entities only required some relatively minor additional computations, we were able to make NER linguistic modifications to the start and end span probabilities locally without the need for a GPU. However, training the model with ELMo word embeddings was quite computationally intensive because it required the model to be re-trained entirely with a new embedding size (1024 for ELMo as opposed to 300 for GloVe). Specifically, the process of running the ELMo embeddings model on each batch to produce the contextualized word embedding representation of each document was computationally intensive.

To account for this, we trained this model using Google Colab's GPU. In order to do this, we parallelized the code using PyTorch's *cuda()* method on our ELMo model. This allowed us to parallelize the process by which contextualized word embeddings were found for each document, giving a huge speedup. Due to resource constraints, we were only able to train for 1 epoch, but even that was sufficient to lead to significant improvements in accuracy.

To retrain our model using the ELMo embeddings, we modified the `--embedding_dim` command line argument and the corresponding model architecture to account for the fact that ELMo has a dimension size of 1024 compared to GloVe's 300.

3 Results

The following table shows the results of our model using the various NER linguistic modification strategies that were applied to the answer spans using the original GloVe embeddings.

	EM	**F1**
Baseline	**20.01**	**31.16**
Named Entity Matching Probability	**19.97**	**31.1**
Named Entity Mapping Probability	**19.8**	**30.54**
Combined Named Entity Probability	**19.75**	**30.5**

The following table shows the results of our model using the ELMo embeddings with and without the combined NER linguistic modification strategy.

	EM	**F1**
GloVe Embeddings	**20.01**	**31.16**
ELMo Embeddings without combined NER strategy	**23.04**	**35.22**
ELMo Embeddings with combined NER strategy	**22.7**	**34.59**

To train our model using the ELMo embeddings, we found that it was infeasible to train directly on our own computers due to resource constraints. As a result, we found that using the Google Colab GPU made drastic improvements to our training time. The most clear and accessible metric to show this improvement was seconds per batch. The table below shows how the training time varied across different resources on training our model with ELMo embeddings.

Device	**Average Seconds per Batch**
2018 Macbook Air, non parallelized	**185**
Google Colab, non parallelized	**94**
Google Colab, parallelized (GPU)	**4**

To access our model, visit our repository at https://github.com/amitjoshi24/QuestionAnswering/

4 Conclusion

4.1 Named Entity Recognition

From these results, we found that our hypothesis that named entities would improve the accuracy was largely incorrect. Cases that we expected like the question of "Who did this?" and the passage of "[X] did this" or "What did [X] buy?" and the passage being "[X] bought apples" were relatively few. Looking into the data, we found that Named Entities from the question will rarely occur in the answer, and many answers may contain no Named Entities in common with the question. For example, if the question was "What is the population of New York?" and the relevant area of the passage was "The Big Apple is home to 8 million people." In these cases, matching named entities would not be of much use, due to the fact that the question and the relevant area of the passage have no named entities in common.

A case where named entity mapping fails is when the question is "What did Abraham Lincoln, who was the 16th president of the United States, write to abolish slavery?" Even though this question contains the word "who," the actual answer ("Emancipation Proclamation") is not a person, organization, group, or country. This type of question, with its multiple question keywords, is too complex for our Named Entity Mapping to capture the type of answer the question is looking for. We found that there are simply too many cases in types and formats of questions for Named Entities to make any significant difference. Perhaps in future research, Named Entity Mapping could be used with a dependency or constituency parser to better know when and when not to map question keywords to Named Entities.

4.2 ELMo Embeddings

We found that using ELMo embeddings in the LSTM drastically improved the model compared to using the original GLoVe embeddings. With GloVe embeddings, the model achieved an EM score of 20.01 and an F1 score of 31.16. With the ELMo embeddings, the model achieved an EM score of 23.04 and an F1 score of 35.22. One factor that leads to these improvements is the fact that ELMo vectorizes each word depending on context. So, in the New York/Big Apple example from earlier, the fact that both of these would have similar contexts helps the model recognize that “Big Apple” means “New York” rather than literally, a big apple (the fruit). This would allow the model to correctly locate the relevant area of the passage where the model is located. Another factor is that ELMo embeddings are simply larger in dimension compared to GLoVe embeddings (1024 vs 300). Larger embeddings allow for more detail to be stored in a single embedding which is useful for increasing accuracy, at the expense of a slower training speed and possibly overfitting (although we didn’t run into this issue).

5 Appendix

The table below shows a complete list of named entity types and their descriptions.

NAMED ENTITY TYPE	**DESCRIPTION**
PERSON	People, including fictional.
NORP	Nationalities or religious or political groups.
FAC	Buildings, airports, highways, bridges, etc.
ORG	Companies, agencies, institutions, etc.
GPE	Countries, cities, states.
LOC	Non-GPE locations, mountain ranges, bodies of water.
PRODUCT	Objects, vehicles, foods, etc. (Not services.)
EVENT	Named hurricanes, battles, wars, sports events, etc.
WORK_OF_ART	Titles of books, songs, etc.
LAW	Named documents made into laws.
LANGUAGE	Any named language.
DATE	Absolute or relative dates or periods.
TIME	Times smaller than a day.
PERCENT	Percentage, including ”%“.
MONEY	Monetary values, including unit.
QUANTITY	Measurements, as of weight or distance.
ORDINAL	“first”, “second”, etc.
CARDINAL	Numerals that do not fall under another type.

6 Acknowledgements

We would like to thank Dr. Durrett and the course staff for an engaging and fruitful semester.

7 References

Danqi Chen, Adam Fisch, Jason Weston, and Antoine Bordes. 2017. *Reading wikipedia to answer opendomain questions.* CoRR, *(https://arxiv.org/pdf/1704.00051.pdf)*.

Diego Mollá, Menno van Zaanen, Daniel Smith. *Named Entity Recognition for Question Answering*. Centre for Language Technology, Macquarie University. (*https://www.aclweb.org/anthology/U06-1009.pdf*)

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, Luke Zettlemoyer. *Deep contextualized word representations*. Allen Institute for Artificial Intelligence, Paul G. Allen School of Computer Science & Engineering, University of Washington. (*https://www.aclweb.org/anthology/N18-1202.pdf*)

Mohit Iyyer, Varun Manjunatha, Jordan Boyd-Graber, Hal Daume III. *Deep Unordered Composition Rivals Syntactic Methods for Text Classification*. University of Maryland, Department of Computer Science and UMIACS. University of Colorado, Department of Computer Science. (*https://people.cs.umass.edu/~miyyer/pubs/2015_acl_dan.pdf*)

spaCy. Named Entity Recognition. (*https://spacy.io/api/annotation#named-entities*)